

# 6.S890: Topics in Multiagent Learning

Lecture 1

Fall 2024

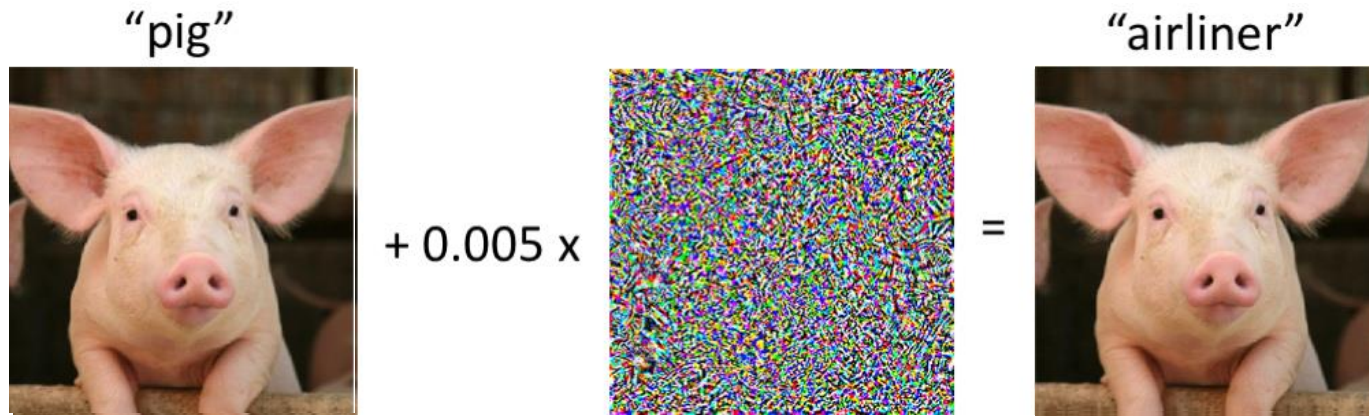


Why treat multi-agent settings differently than single-agents?

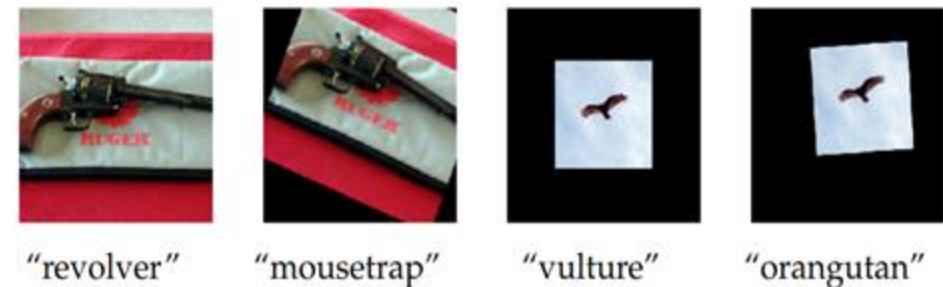
(I) Strategic Behavior does not emerge from standard training



# (II) Naively trained models can be manipulated



[Athalye, Engstrom, Ilyas, Kwok ICML'18]



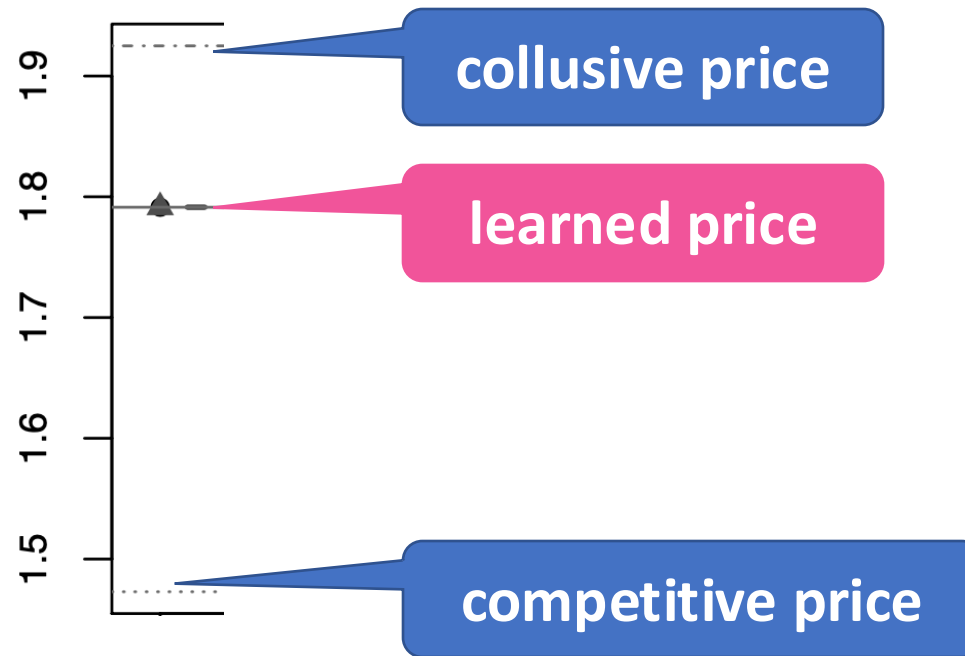
[Engstrom et al. 2019]

# (III) Combining agents that were trained in isolation can lead to undesirable behavior

**Example:** AI for dynamic pricing

**Setting:** Duopoly w/ two symmetric firms

**Independent Learning:** firms cannot communicate other than setting prices, observing their profit and adjusting their price using some standard AI algorithm



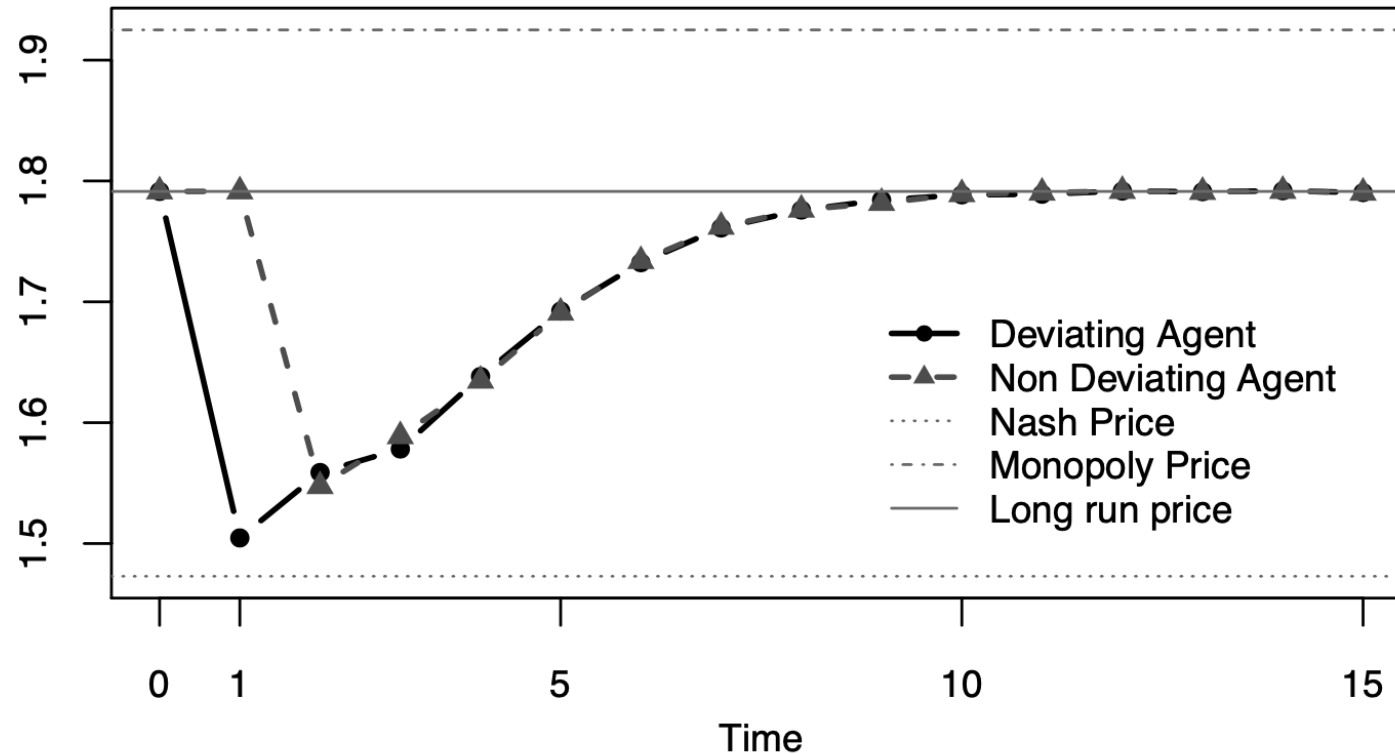
[Calvano, Calzolari, Denicolo, Pastorello: "Artificial Intelligence, Algorithmic Pricing, and Collusion," American Economic Review, 2020]

# (III) Combining agents that were trained in isolation can lead to undesirable behavior

**Example:** AI for dynamic pricing

**Setting:** Duopoly w/ two symmetric firms

**Independent Learning:** firms cannot communicate other than setting prices, observing their profit and adjusting their price using some standard AI algorithm



How deviations are punished by the learned price policies

[Calvano, Calzolari, Denicolo, Pastorello: "Artificial Intelligence, Algorithmic Pricing, and Collusion," American Economic Review, 2020]

(IV) The optimization workhorse of Deep Learning struggles in multi-agent settings

# (IV) The optimization workhorse of Deep Learning struggles in multi-agent settings

$$\min_{\theta} \ell(\theta)$$

$\theta$ : high-dimensional

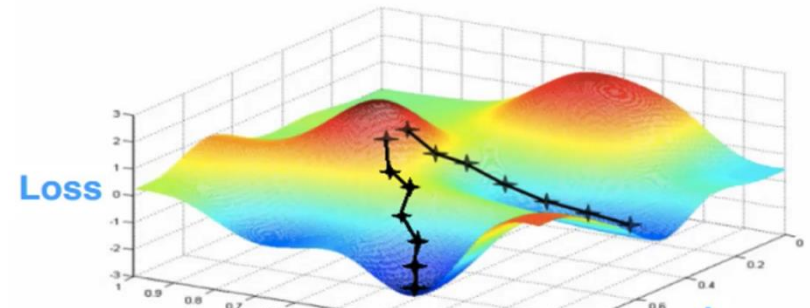
$\ell$ : **nonconvex**

essentially only accessible through  $\ell(\theta)$  and  $\nabla \ell(\theta)$  queries

STANDARD DEEP LEARNING ESTIMATION PROBLEM

$$\theta_{t+1} = \theta_t - \eta \cdot \nabla \ell(\theta_t)$$

Gradient Descent



**Theoretical Guarantee:** Even if  $\ell$  **nonconvex**, Gradient Descent efficiently computes *local minima*

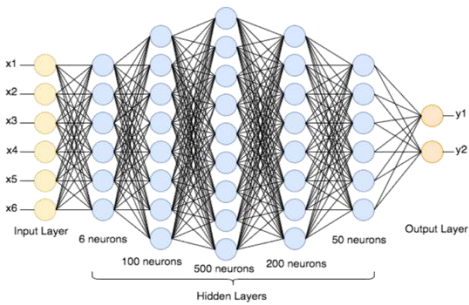
*[Lee et al 2017, Ge et al '15]*

**Empirical Finding:** *Local minima* are good enough



# (IV) The optimization workhorse of Deep Learning struggles in multi-agent settings

Prominent Paradigm:



$$+ \theta_{t+1} \leftarrow \theta_t - \nabla_{\theta} \ell(\theta_t)$$

+



+



Caffe

Caffe2

Chainer

Microsoft  
Cognitive  
Toolkit

MATLAB

mxnet

PaddlePaddle

PyTorch

TensorFlow

torch

Wolfram  
Language

# (IV) The optimization workhorse of Deep Learning struggles in multi-agent settings

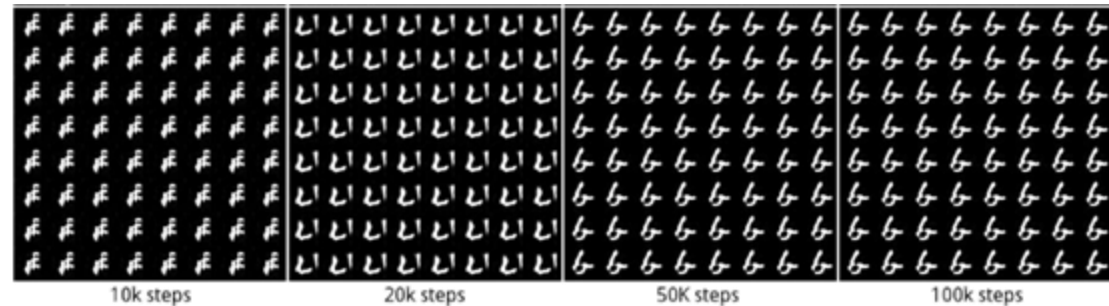
GANs:  $\ell_D(\theta, \omega) = -\ell_G(\theta, \omega)$   
 $\ell_G, \ell_D$ : nonconvex in  $\theta$  &  $\omega$  resp.;  
 $\theta, \omega$ : high-dimensional

Simultaneous Gradient Descent (GD) Dynamics:

$$\theta_{t+1} = \theta_t - \eta \cdot \nabla_{\theta} \ell_G(\theta_t, \omega_t)$$
$$\omega_{t+1} = \omega_t - \eta \cdot \nabla_{\omega} \ell_D(\theta_t, \omega_t)$$

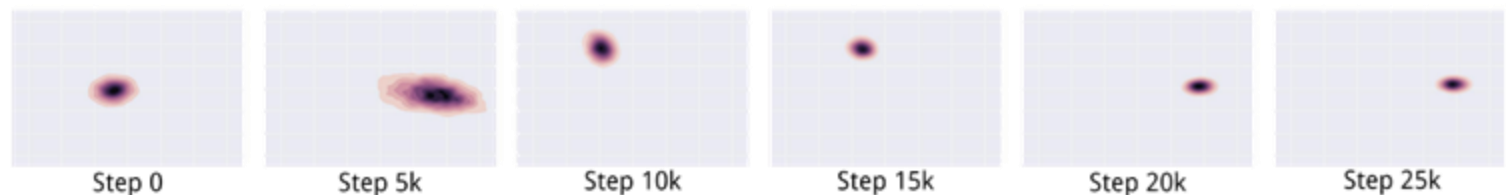
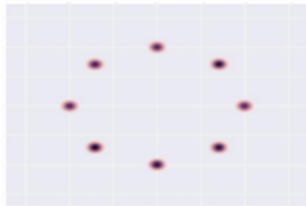
GAN training on MNIST Data:

Target:



GAN training on Gaussian Mixture Data:

Target:



pictures from [Metz et al ICLR'17]

# This course

**How can we, and machines, systematically reason about the behavior, incentives, and outcomes of multiagent systems?**

And as computer scientists, how can we teach machines to compute, predict, or learn such behavior?

# Two pervasive concepts

- **Equilibrium**: a situation in which no player has incentives to change their policy
  - Different types exist depending on what the players can communicate, observe, etc.
  - Mental picture: multi-agent generalization of the concept of local optimum in optimization
  - Typically being an equilibrium is a *necessary* condition for what you would consider a “solution” or “optimal strategy” in a game
- **Learning in games**: global equilibrium can be reached from local improvement steps

## Learning in games

*Constructive* answer to the following natural question:

“Can a player that repeatedly plays a game follow rules to refine their strategy after each match, so as to guarantee *mastering* the game in the long run?”

Today learning techniques are typically the fastest way to compute high-quality solutions for large strategic interactions

# Course goals

- By the end of this part, you should have acquired:
  - A **language** to think about and describe different equilibrium points of multiagent interactions (Nash equilibrium, maxmin strategies, correlated equilibria, ...)
  - An appreciation for what is **computationally tractable** in every case, and what only in special cases
  - The ability to **implement learning dynamics** to progressively refine strategies, including in imperfect-information domains
  - A general understanding of what techniques are used to push scalability, and what major areas of investigation remain **underexplored**

# A taste of the computational challenges

- By the end of the course, you should have a much clearer picture of how you could model and solve the following tasks computationally

## 1. Bluffing in poker

How can we mathematically quantify the optimal amount of bluffing?

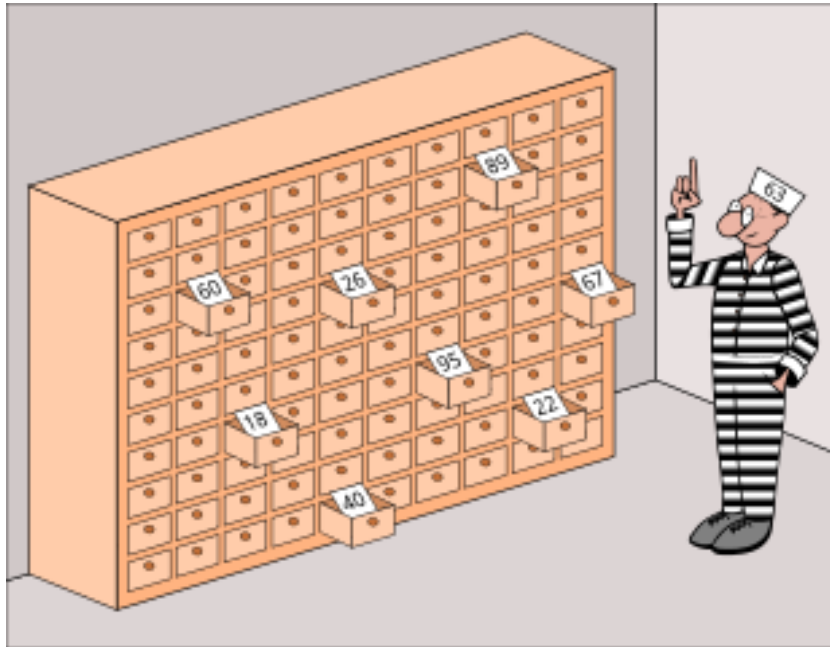
Note that bluffing contradicts the idea that the optimal strategy can be computed inductively by traversing the game tree

Principle in imperfect-information games: need to be careful not to “leak secrets”.

## 2. The value of cooperation

How do we model and solve a cooperative but imperfect-information game like Hanabi?

How do we model and solve the following situation (described in a paper co-authored by game theorist Peter Bro Miltersen)?



*The director of a prison offers 100 death row prisoners, who are numbered from 1 to 100, a last chance. A room contains a cupboard with 100 drawers. The director randomly puts one prisoner's number in each closed drawer. The prisoners enter the room, one after another. Each prisoner may open and look into 50 drawers in any order. The drawers are closed again afterwards. If, during this search, every prisoner finds their number in one of the drawers, all prisoners are pardoned. If even one prisoner does not find their number, all prisoners die. Before the first prisoner enters the room, the prisoners may discuss strategy — but may not communicate once the first prisoner enters to look in the drawers. What is the prisoners' best strategy?*



# Prisoners puzzle

- Single agent solution: look at random -> 50% win probability
- With 100 prisoners:
  - Since the prisoners cannot communicate,  $(50\%)^{100} = 10^{-30}$  win probability?
  - **Turns out there is a strategy with 30% win probability**
  - Remember: players cannot communicate during the game. If even one player fails, everyone fails.
- The important question: Computationally, how would you approach this strategy computation problem?
- What about variants: for example, an adversarial prison guard might swap two drawers. How does the value of the game change?

### 3. Computation of optimal mechanisms

How can we model the task of computing an optimal auction mechanism as an imperfect-information game? We want to make sure that no player would be better off by misreporting their evaluation of an item

- This is a game between the player and the mechanism
- Like all the other examples, it can be reformulated as an optimization problem
- It can also be solved using learning techniques, like the other settings

# The Concept of Game

*Games* are thought experiments to help us learn how to *predict rational behavior* in *situations of conflict*.

*Situation of conflict:* Everybody's actions affect others.

*Rational Behavior:* The players want to maximize their own expected utility. No altruism, envy, masochism, or externalities (if my neighbor gets the money, he will buy louder stereo, so I will hurt a little myself...).

*Predict:* We want to know what happens in a game. Such predictions are called *solution concepts* (e.g., Nash equilibrium).

# Situations Modeled as Games

- **Recreational** games

- Rock paper scissors
- Diplomacy
- Poker
- Go
- ...

- **Non-recreational** settings

- Auctions
- Markets
- Logistics
- Budget allocation (e.g., political campaigns)
- Generative networks
- Multi-robot interactions
- Fraud detection systems
- ...

# Administrivia

- The course will use **Canvas**
- Lecturers: Gabriele Farina
- TA: Zhiyuan Fan
- Attendance: Everyone is welcome! If *auditing* please register as a listener
- Office hours: TBD

# Projects

- **Problem sets:** 3 problems sets, two weeks to solve each (weight: 50%)
- **Project: proposal** due October 18<sup>th</sup> (weight: 50%)
  - Project **brainstorming** class on 10/10
  - Project **break** on the week of 10/22-24 and 11/26-28
  - Project **presentations** starting 12/3
  - Project can be theoretical, practical, or a mix
  - We encourage creativity!
  - Feel free to run your ideas by us
  - Feel free to apply ideas from this class to your own area of interest
  - Project can be done in teams
- **No exams**

# Project type #1: Wildcard

- You pick the project
- It can be theoretical or empirical
- We will give out some ideas to get you started, but feel free to propose anything else

# Project type #2: Tic-Tac-Toe Competition

- OK not that easy. It will be an **imperfect-information** variant: there is a **fog of war** on the board
- You can pick a cell
  - If the cell is empty, you are in luck: place your mark there
  - If the cell is occupied: the move will fail (you will observe that) and the turn passes to your opponent.
- Goal: Compute an optimal (maxmin) strategy for this game
- We will keep a leaderboard of attempts and results (But don't worry, your grade depends on the final report, it is not a problem to end up last as long as what you tried made sense!)



# Project type #2: Tic-Tac-Toe Competition

- You can pick any method you want:
  - Deep RL
  - Tabular no-regret learning algorithms
  - Convex optimization techniques
  - ...
- You will produce a policy for the game and we will compute the expected value and exploitability
- Your report should explain what you tried and what you found

# Project type #3: Team poker

- This project is about exploring games with a mix between cooperative and competitive aspects
- We will use a small 4-player poker variant
  - You will control a team of players facing each other
  - The payoff of the team is the sum of the players' returns in the game
  - The players cannot communicate during the game, but they can discuss any tactics before the game begins, which allows some form of “tacit collusion”
- Similar mechanics as the previous project

# Tentative schedule





1	Sep 5 <sup>th</sup>	<b>Introduction</b>
PART I: NORMAL-FORM GAMES		
2	Sep 10 <sup>th</sup>	<b>Setting and equilibria: Nash equilibrium</b> Definition of normal-form games. Solution concepts and Nash equilibrium. Nash equilibrium existence theorem. Brouwer's fixed point theorem. von Neumann's minimax theorem.
3	Sep 12 <sup>th</sup>	<b>Setting and equilibria: Correlated equilibrium</b> Definition of correlated and coarse correlated equilibria. Their relationships with Nash equilibria. Linear programming formulations of equilibrium.

# Part I: Normal-Form Games

*These are “matrix” games*

- Simultaneous actions
- Single move per player

Simple model but already captures several important aspects

		Rock 	Paper 	Scissors 
Rock 		0,0	-1,1	1,-1
Paper 		1,-1	0,0	-1,1
Scissors 		-1,1	1,-1	0,0

Rock-paper-scissors

		Deny (cooperate)	Confess (betray)
Deny (cooperate)		-1, -1	-3, 0
Confess (betray)		0, -3	-2, -2

Prisoner's dilemma

(-1 = 1 year in jail)

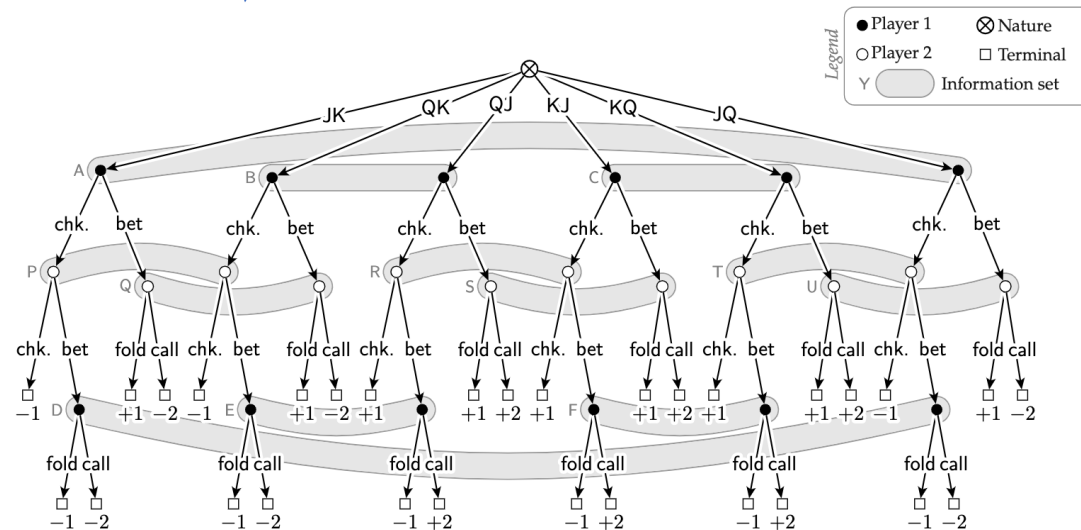
4	Sep 17 <sup>th</sup> <b>HW1 OUT</b>	<b>Learning in games: Foundations</b> Regret and hindsight rationality. Definition of regret minimization and relationships with equilibrium concepts.
5	Sep 19 <sup>th</sup>	<b>Learning in games: Algorithms (part I)</b> General principles in the design of learning algorithms. Follow-the-leader, regret matching, multiplicative weights update, online mirror descent.
6	Sep 24 <sup>th</sup>	<b>Learning in games: Algorithms (part II)</b> Optimistic mirror descent and optimistic follow-the-regularized-leader. Accelerated computation of approximate equilibria.
7	Sep 26 <sup>th</sup>	<b>Learning in games: Bandit feedback</b> From multiplicative weights to EXP3. General principles. Obtaining high-probability bounds.
8	Oct 1 <sup>th</sup>	<b>Learning in games: <math>\Phi</math>-regret minimization</b> Gordon, Greenwald, and Marks (2008); Blum and Mansour; Stolz-Lugosi.

Despite their simplicity, **normal-form games will provide the ground** to start looking into the following key concepts in multiagent settings:

- Solution concepts and equilibria (Nash, maxmin, correlated, ...)
- Learning from repeated play
  - Learning enables iteratively refining strategies to become stronger and stronger, and it has been a key component in all recent game AI breakthroughs
  - Local learning of each agent can often be connected to global notion of equilibrium
  - $\approx$  Mental model: “reinforcement learning but also works in nonstationary settings”
- Deep connection between equilibria and other important concepts in computer science
- After that, we will move on to notions of games that capture more interesting / real-world phenomena, especially sequential moves and imperfect information

# Part II: Imperfect-Information Games

Example:  
poker



# Difficulties with Imperfect Information

- Compared to normal-form games, imperfect-information extensive-form games bring many conceptual challenges

1 The number of (deterministic) strategies grows **exponentially** in the game tree

2 Imperfect information makes backward induction and local reasoning not viable

Think about poker: need to reason about **misdirection**. *General principle: you need to think about what the opponents don't know about you and leverage that to your advantage*

3 Other players have control over what part of the game tree is visited/explored

- Nonetheless: many positive results
  - In fact, we live in a world where machines bluff at poker better than humans



## PART II: EXTENSIVE-FORM GAMES

9	Oct 3 <sup>rd</sup>	<b>Foundations of extensive-form games</b> Complete versus imperfect information. Kuhn's theorem. Normal-form and sequence-form strategies. Similarities and differences with normal-form games.
10	Oct 8 <sup>th</sup> <b>HW2 OUT</b>	<b>Learning in extensive-form games</b> Counterfactual regret and counterfactual regret minimization (CFR). Proof of correctness and convergence speed.
11	Oct 10 <sup>th</sup>	<b>PROJECT</b> Project ideas and brainstorming
12	Oct 15 <sup>th</sup>	<b>HOLIDAY</b> No class (Student holiday)
13	Oct 17 <sup>th</sup>	<b>Equilibrium refinements</b> Sequential irrationality. Extensive-form perfect equilibria and quasi-perfect equilibrium.

14	Oct 22 <sup>nd</sup>	<b>PROJECT</b> Project break Coincides with INFORMS 2024 Annual Meeting
15	Oct 24 <sup>th</sup>	<b>PROJECT</b> Project break Coincides with INFORMS 2024 Annual Meeting
16	Oct 29 <sup>th</sup>	<b>Deep reinforcement learning for large-scale games (part I)</b> Rough taxonomy of deep RL methods for games. Decision-time planning in imperfect-information games, construction of superhuman agents for no-limit Hold'em poker. Public belief states techniques (ReBeL).
17	Oct 31 <sup>st</sup>	<b>Deep reinforcement learning for large-scale games (part II)</b> PPO and magnetic mirror descent.

# **Part III: Other structures**

## PART III: OTHER STRUCTURED GAMES

18	Nov 5 <sup>th</sup>	<b>Succinct games and computation of exact equilibria</b> Example of succinct games. Ellipsoid against hope algorithm. Generalization of ellipsoid against hope to other equilibrium concepts.
19	Nov 7 <sup>th</sup>	<b>Combinatorial games</b> Example of combinatorial games. Kernelized multiplicative weights update algorithm.
20	Nov 12 <sup>th</sup> <b>HW3 OUT</b>	<b>Polymatrix and potential games</b> Definitions, taxonomy, and examples (congestion games, network games, atomic games). Equilibrium computation in potential games.
21	Nov 14 <sup>th</sup>	<b>Stochastic games</b> Minimax theorem, and existence of equilibrium. Stationary Markov Nash equilibria.

# **Part IV: Complexity of equilibrium**

# Nash's Theorem

**[John Nash '50]**: A Nash equilibrium exists in every finite game.

Deep influence in Economics, enabling other existence results.

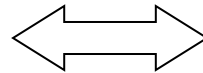
Proof highly non-constructive (uses Brouwer's fixed point thm)

No simpler proof has been discovered

**[Daskalakis-Goldberg-Papadimitriou'06]**: no simpler proof exists

i.e.

**Nash  
Equilibrium**



**Brouwer's Fixed  
Point Theorem**

PART IV: COMPLEXITY OF EQUILIBRIUM COMPUTATION

22	Nov 19 <sup>th</sup>	<b>PPAD-completeness of Nash equilibria (part I)</b> Sperner's lemma. The PPAD complexity class. Nash $\in$ PPAD.
23	Nov 21 <sup>st</sup>	<b>PPAD-completeness of Nash equilibria (part II)</b> Arithmetic circuit SAT. PPAD-hardness of Nash equilibria.

# And finally...

## PROJECT BREAK & PRESENTATIONS

24	Nov 26 <sup>th</sup>	<b>PROJECT</b> Project break
25	Nov 28 <sup>th</sup>	<b>HOLIDAY</b> No class (Thanksgiving)
26	Dec 3 <sup>rd</sup>	<b>PROJECT</b> Project presentations
27	Dec 5 <sup>th</sup>	<b>PROJECT</b> Project presentations



# Course goals (again)

- By the end of this part, you should have acquired:
  - A **language** to think about and describe different equilibrium points of multiagent interactions (Nash equilibrium, maxmin strategies, correlated equilibria, ...)
  - An appreciation for what is **computationally tractable** in every case, and what only in special cases
  - The ability to **implement learning dynamics** to progressively refine strategies, including in imperfect-information domains
  - A general understanding of what techniques are used to push scalability, and what major areas of investigation remain **underexplored**

Questions?